

# Dynamical Adaptation in ORNs

Srinivas Gorur-Shandilya

January 21, 2014

## Dynamical Adaptation in ORNs

Do ORNs exhibit fast adaptation to a flickering stimulus? Can a simple dynamical adaptation model predict ORN responses to flickering inputs? Here, I take data from Carlotta's random stimulus experiments and first check how well a simple linear prediction from the stimulus compares to the data. Then, I study how the instantaneous gain of the actual ORN response w.r.t to the predicted ORN response varies with time, and try to find a objective filter from the stimulus to this instantaneous gain to correct for systematic errors in the linear prediction.

## Contents

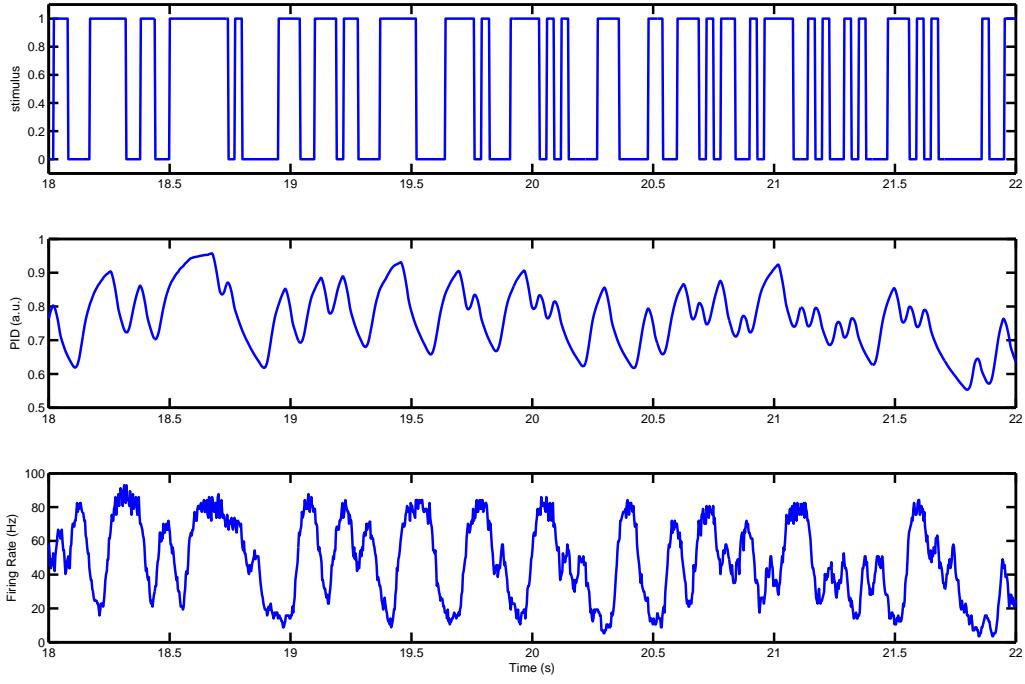
- Data Visualisation
- Filter Extraction
- Analysis of Linear Prediction - Linearity of Prediction
- Analysis of Linear Prediction - Which input predicts output better?
- Analysis of Linear Prediction - Response to High and Low Stimuli
- Analysis of Linear Prediction - Filter Variation
- Analysis of Linear Prediction - Instantaneous Gain
- Summary/Problems
- Next Steps

## Data Visualisation

Data from this file will be used for this analysis.

`/data/random-stim/final_2011_06_14_ab3A_1o3o13X-3_20ml_30sec_30ms_rand.mat`

Here, data from simultaneous measurements of the stimulus and from the ORN is shown. The top row shows the valve command signal (a binary signal, high means valve is open). The middle row shows the simultaneous measurement of the odour stimulus with a PID, and the bottom row shows the instantaneous firing rate of the ORN recorded from.



The odour used, the neuron recorded from and the correlation time of the flickering stimulus are in the file name displayed above the plot.

## Filter Extraction

Now, we extract a simple linear filter from the stimulus such that a convolution with the input (stimulus) gives the output (ORN firing rate).

The filter is calculated using Chichilnisky's method. The only modification I have used is to regularise the matrix before inversion. The code is based on what Carlotta uses, and I have re-written it for clarity and modified the regularisation step.

The filter  $K$  is given by

$$C * K = s' * f$$

where  $C$  is

$$C = Cov(s) + \frac{rI}{1+r}$$

where  $I$  is the identity matrix and  $r$  is a free parameter called the regularisation factor that suppresses the high-frequency components of  $K$ .  $s$  is the stimulus vector (e.g. the PID) and  $f$  is the response vector

(here, the firing rate of the ORN). In practice,  $K$  is estimated by a left matrix division:

$$K = C \setminus (s' * f)$$

For comparison, I have also calculated the filter using Damon's function (Back\_out\_1dfilter\_new)

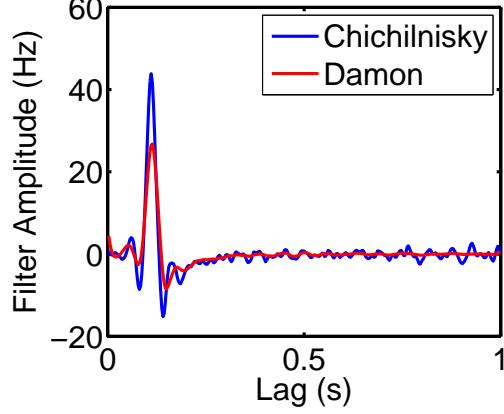
The problem of choosing  $r$  appropriately remains. If  $r$  is too low, then the effective number of parameters of the filter goes up, permitting over-fitting of the data, and you end up with a noisy squiggle of a filter that fits the data well. If  $r$  is too high, the filter has too few modes, and doesn't fit the data well, but looks nice. The ideal  $r$  is chosen so that it is equal to

$$\arg \min_r \left\| \frac{e(r) - \min(e)}{\min(e)}, \frac{S(r) - \min(S)}{\min(S)}, \frac{|m(r) - 1|}{1} \right\|_{\infty}$$

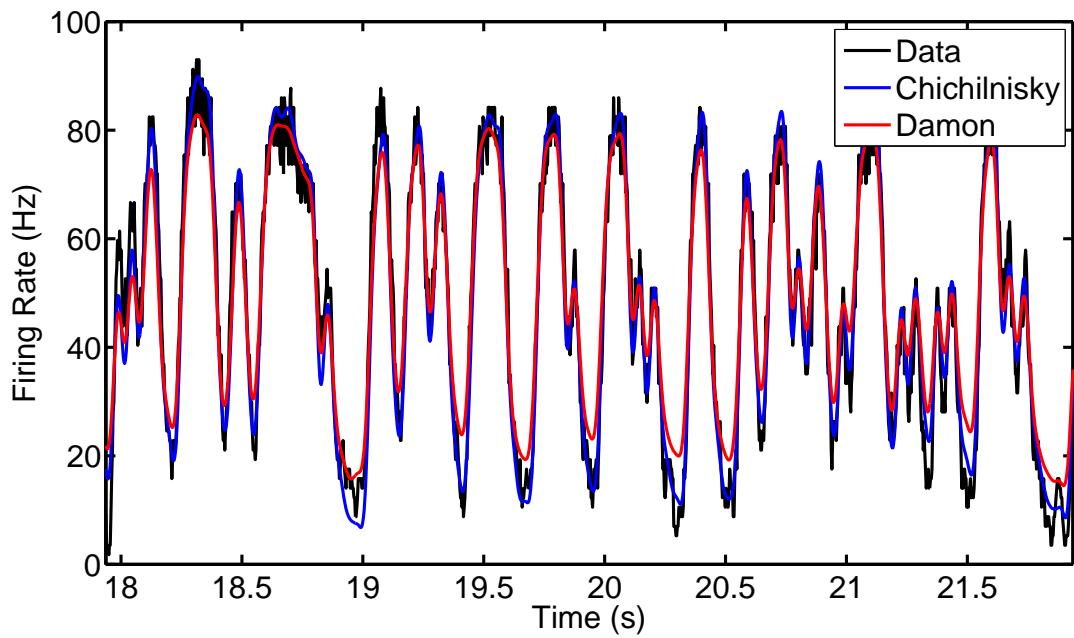
where  $e$  is the error of the prediction w.r.t to the actual data,  $S$  is the absolute sum of the filter  $K$ , and  $m$  is the slope of the best fit of the prediction to the data. Minimising  $e$  minimises the error of the fit. Minimising  $S$  minimises the high-frequency components in the filter, and prevents over-fitting. We also want the slope  $m$  to be as close to 1 as possible.

For a detailed look at how the slope of best fit and the filter shape vary with  $r$ , look at the next section.

In the figure below, the filter computed the PID and the firing rate is shown. In blue is the filter calculated using Chichilnisky's method, and in red is the filter calculated using Damon's function. The x axis is the lag of the filter, and the Y axis is the amplitude of the filter (in Hz)

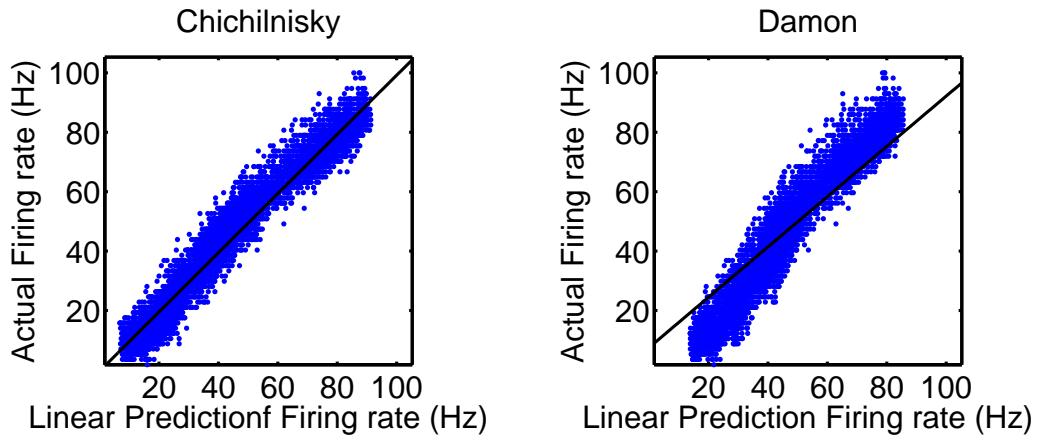


The convolution of this filter with the input (PID) gives a prediction of the output. In the figure below, the data is shown in black, and the prediction from this linear filter is shown in blue (Chichilnisky method) and red (Damon's function). Note that sometimes the prediction is below zero, which doesn't mean anything physical (firing rate cannot be < 0).



### Analysis of Linear Prediction - Linearity of Prediction

For a linear filter calculated from the data, a plot of the actual response to the predicted response can be fit with a line of slope 1. Here the actual firing rate is plotted on the X axis and the firing rate predicted from the linear filter is plotted on the Y axis.



The line is a best fit to all the data. The slope of the best fit line to Chichilnisky's fit is:

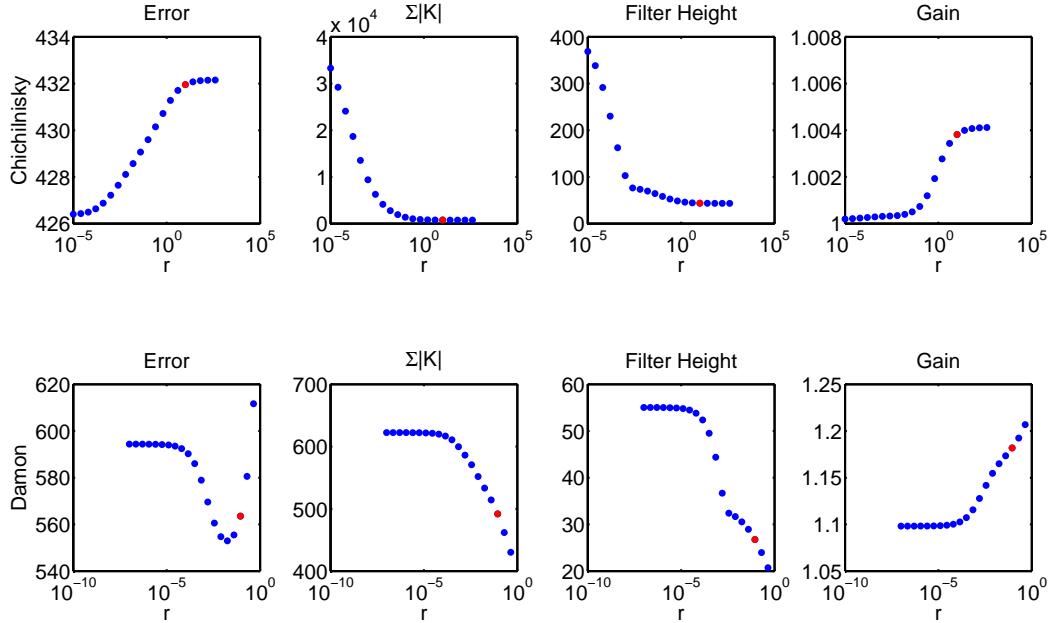
1.0038

The line is a best fit to all the data. The slope of the best fit line to Damon's fit is:

1.1819

Note that for the Chichilnisky filter, the slope is very close to 1. For very little regularisation, the slope is exactly 1.

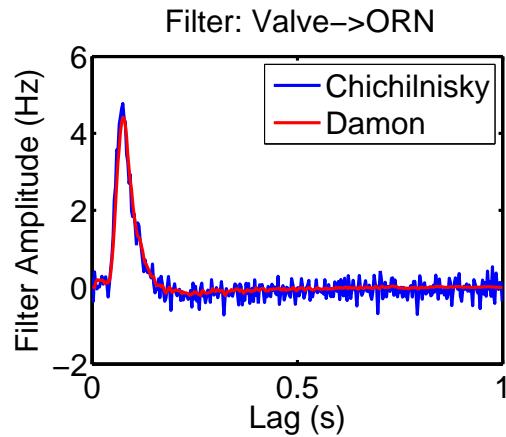
Both Damon's and my filter estimation functions have a free parameter,  $r$ , the regularisation factor. The following plots show how the error, the filter sum, and the filter height vary with varying the free parameter  $r$ . The point in red indicates the value of  $r$  finally chosen for the best filter.



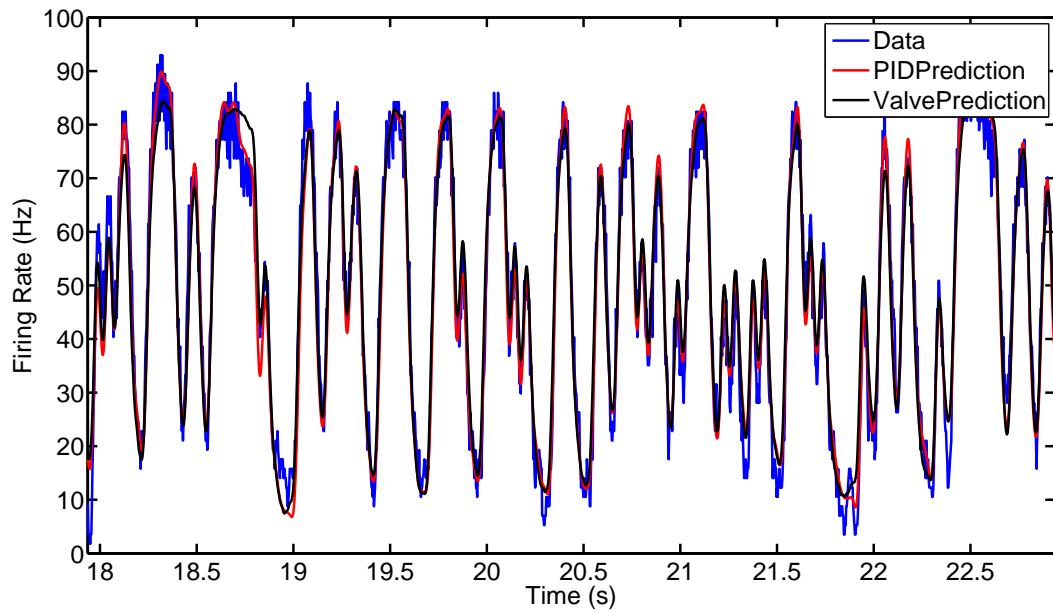
### Analysis of Linear Prediction - Which input predicts output better?

The measurement of the stimulus should help us predict the response of the neuron better than just the valve control signal, as we now capture the fast fluctuations of the odour stimulus as it reaches the ORN. Thus, a prediction of ORN firing response from the PID signal should be better than a prediction of the ORN firing response from the valve signal.

First, we recalculate a new filter from the valve control signal and the ORN firing rate using the same methods as before. The plot below shows this new filter:



Using the filter calculated using Damon's code, we make a new prediction of the firing rate of the ORN. In the figure below, the data is shown in blue, the linear prediction from the PID is shown again in red, and the linear prediction from the Valve is shown in black.



The quality of the prediction is measured using the Root-mean-square error, calculated as:

$$\Delta t * \sqrt{\sum (f - f_p)^2}$$

The error in the prediction from the PID, per unit time is:

1.3014

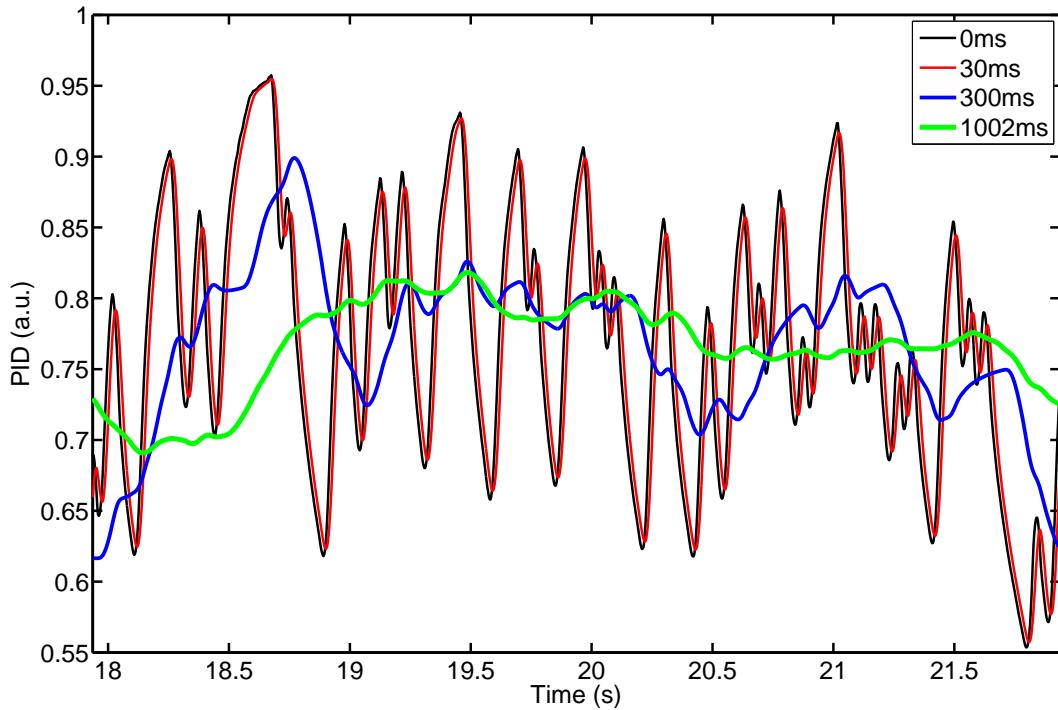
while the error for the prediction from the Valve is:

1.3955

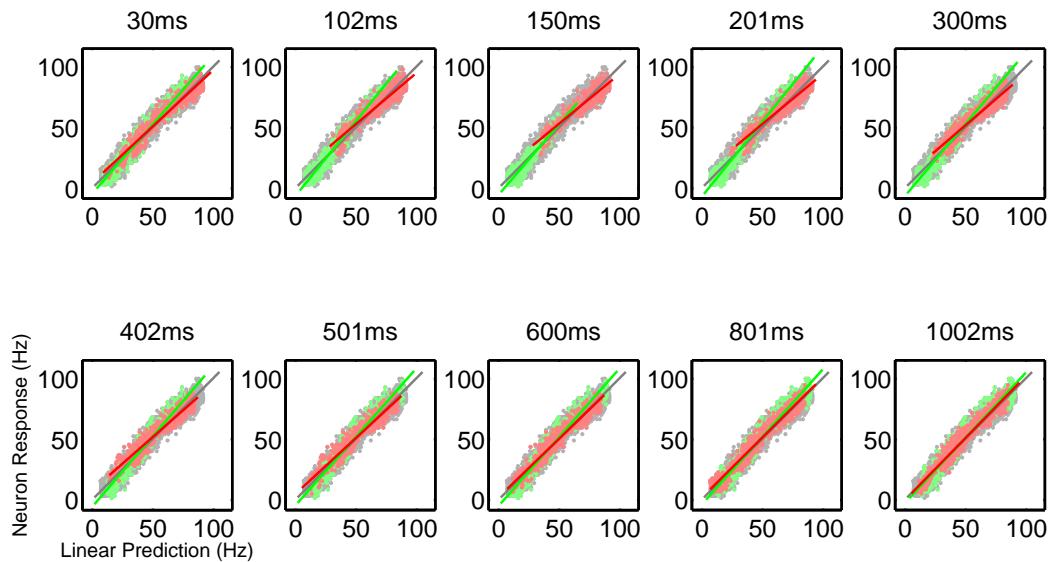
## Analysis of Linear Prediction - Response to High and Low Stimuli

How does the response of the ORN differ for high and low stimuli? Specifically, does the neuron display the characteristics of fast adaptation to this flickering stimulus?

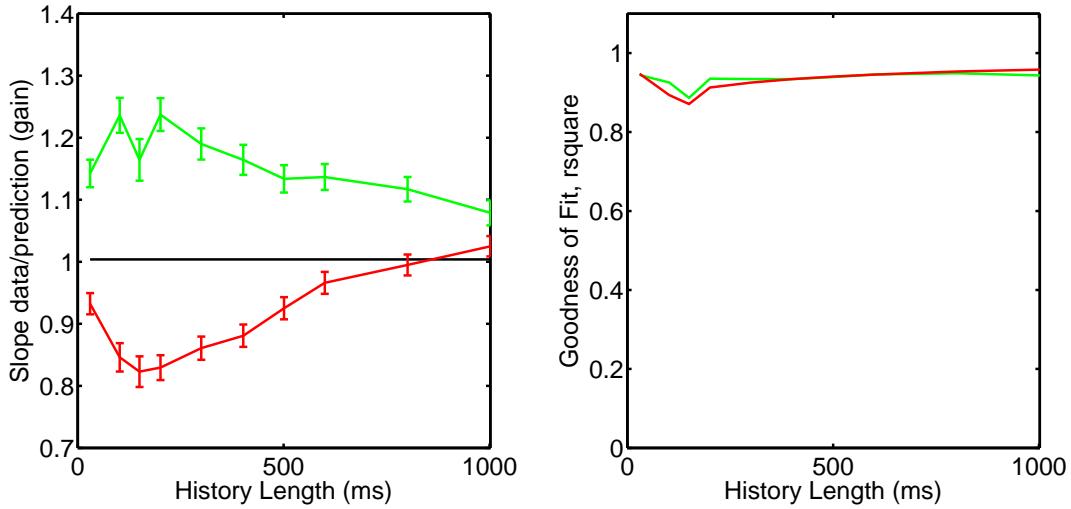
To look at this, we calculate the mean average stimulus over some window history length for every time point  $t$ , for various different window history lengths. The effect of this operation is shown in the figure below, where the legend refers to the history window in ms.



Now, we separate neuron responses at times when the mean average stimulus is in the lowest 10% or the highest 10%. These points are marked either green (lowest 10%) or red (or highest 10%) in the figures below, while all the data is plotted in grey. Lines are fit to each of these clouds of points, and the slopes (representing the instantaneous gain) is calculated from these lines. The  $y$  axis is the actual firing rate, while the  $x$  axis is the predicted firing rate.



The following plot shows how the slope of the lines of best fit, or the instantaneous gains, varies with the history length. The plot on the right shows the goodness of fit for each fit, indicating the regions where the fit is meaningful.

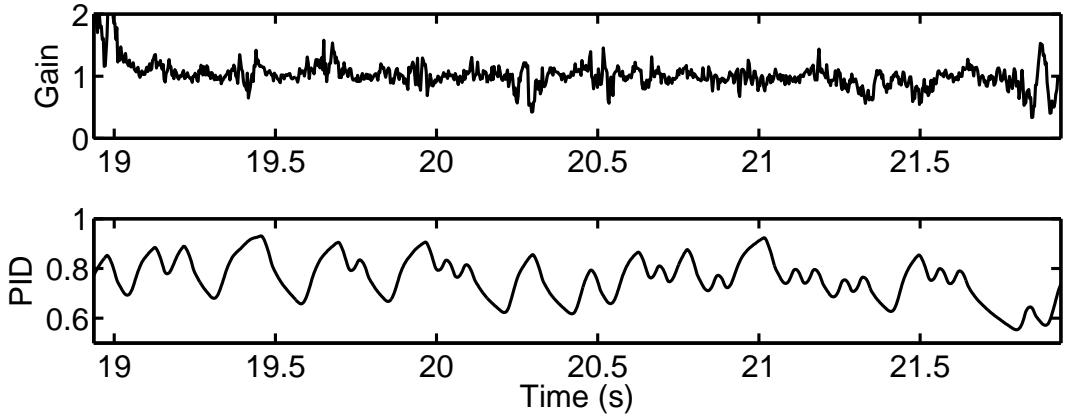


### Analysis of Linear Prediction - Filter Variation

We have segmented the entire data set based on when the stimulus, averaged in some way over the past, into high and low. Here, I will calculate filters for each of these subsets.

### Analysis of Linear Prediction - Instantaneous Gain

There is a mismatch between the linear prediction and the actual firing rate. Moreover, the instantaneous gain seems to be modulated by something that depends on the past history of the stimulus. Here, in the figure below, the instantaneous gain, i.e., the ratio of the actual firing rate to the predicted firing rate, is plotted along with the stimulus.

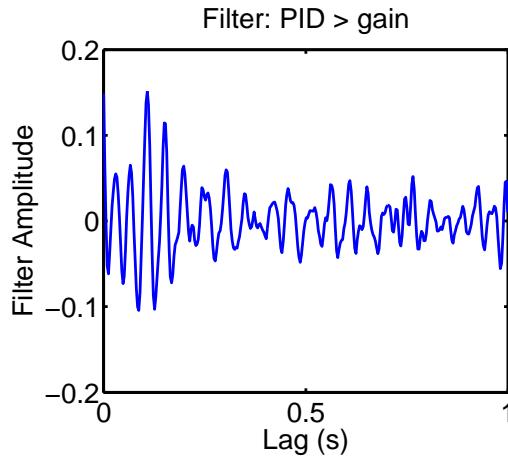


In the above analysis, we have considered how a boxcar average over the stimulus immediately preceding the current time affects gain. Now, we want to find some optimal way of averaging the past stimulus history to predict the instantaneous gain: by doing so, we can then predict instantaneous gain, and thus get a better predictor of the actual firing rate.

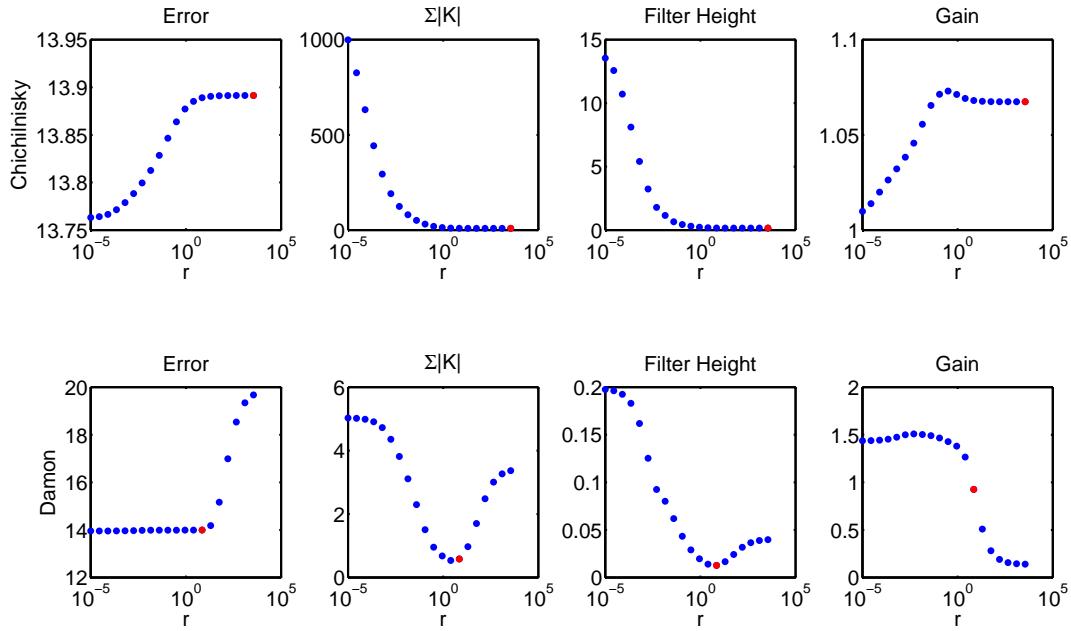
In effect, we can calculate a new filter,  $K_g$  such that

$$K_g = C \setminus (s' * g)$$

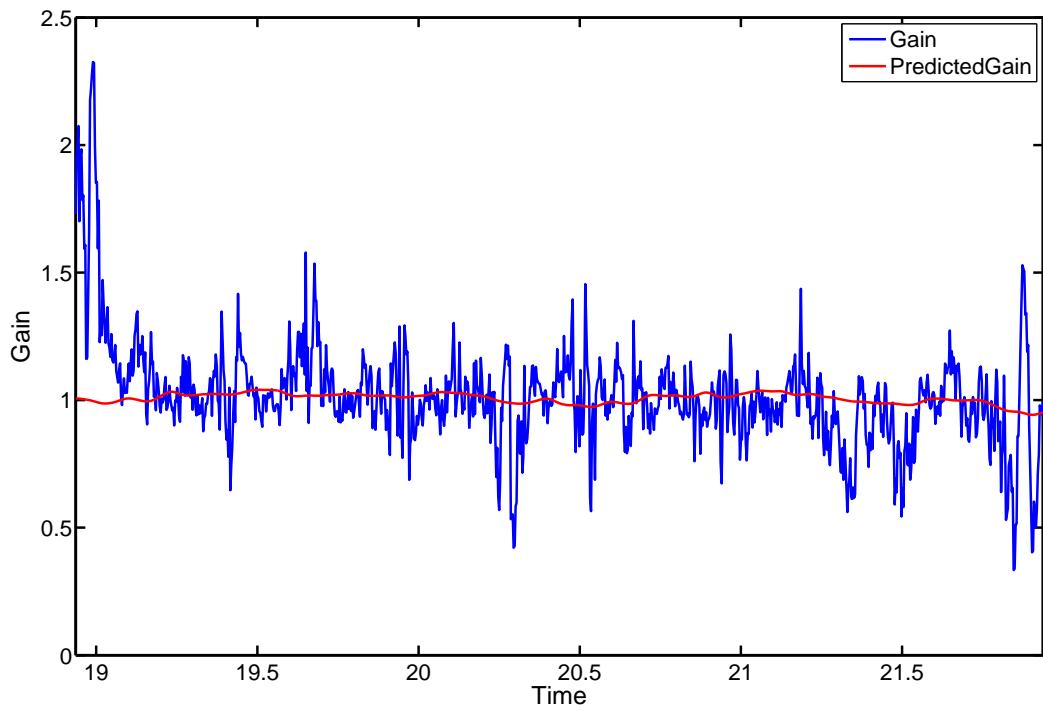
where  $g$  is the instantaneous gain and  $s$  is the stimulus.



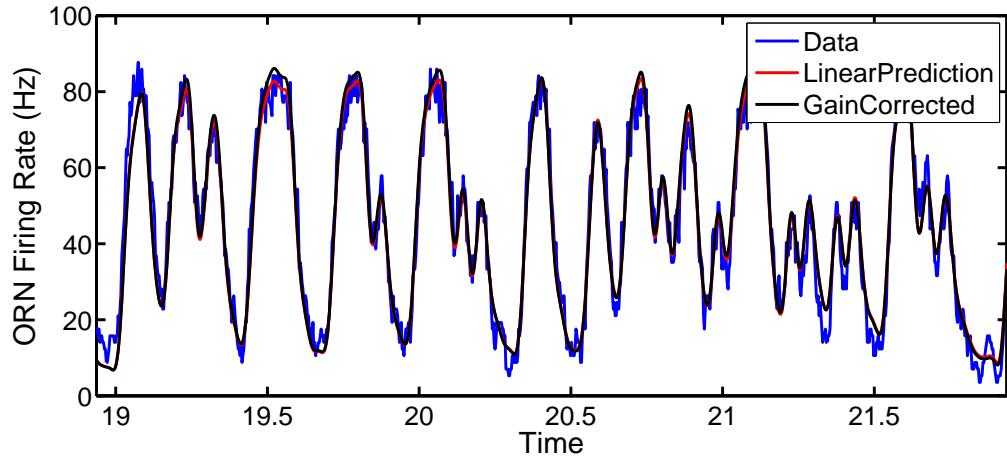
Once again, we can vary the free parameter to see how to best choose a filter.



From this filter and the stimulus, we can estimate the instantaneous gain (where the instantaneous gain is defined to be the ratio of the actual firing rate to the linear predicted firing rate)



Using this estimate, we can then correct the linear prediction of the firing rate.



Is the gain-corrected prediction any better than the linear prediction? The mean error of the gain-corrected prediction is

1.4087

while the mean error of the simple linear prediction is

1.3014

No matter what I do, I can't get use a linear filter to predict the gain well from the stimulus.

## Summary/Problems

1. Best fit line of linear prediction to actual data does not have a slope of 1 for Damon's function. For synthetic data the slope of the line is indeed 1. However, in the real data set, it is not, even with no regularisation.
2. Prediction of gain from past stimulus is very poor. The correlation coefficient of the prediction with the instantaneous gain is  $\sim 0.2$ . For comparison, the corr. coeff. of the firing rate prediction  $> 0.95$ .

## Next Steps

1. Calculate filters for high and low stimulus zones.
2. Calculate filters after using elastic net regulararisation.
3. Check if someone has looked at Weber's law in olfaction/ORNs
4. shuffle data to check validity?
5. fit Dynamical Adaptation model to data?